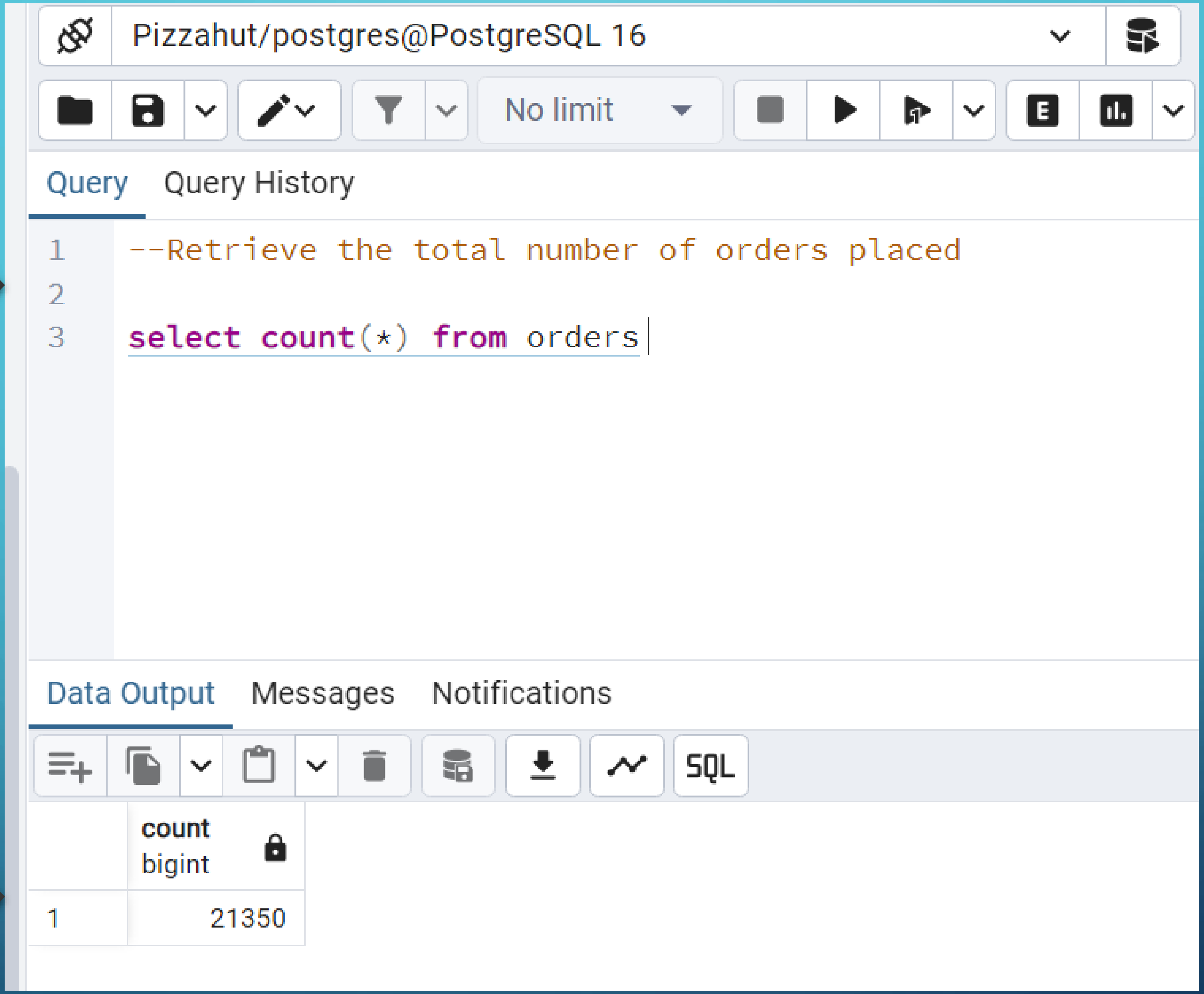
DATA MASTERY WITH SQL: CRAFTING EFFICIENT QUERIES

INPUT QUERY

OUTPUT

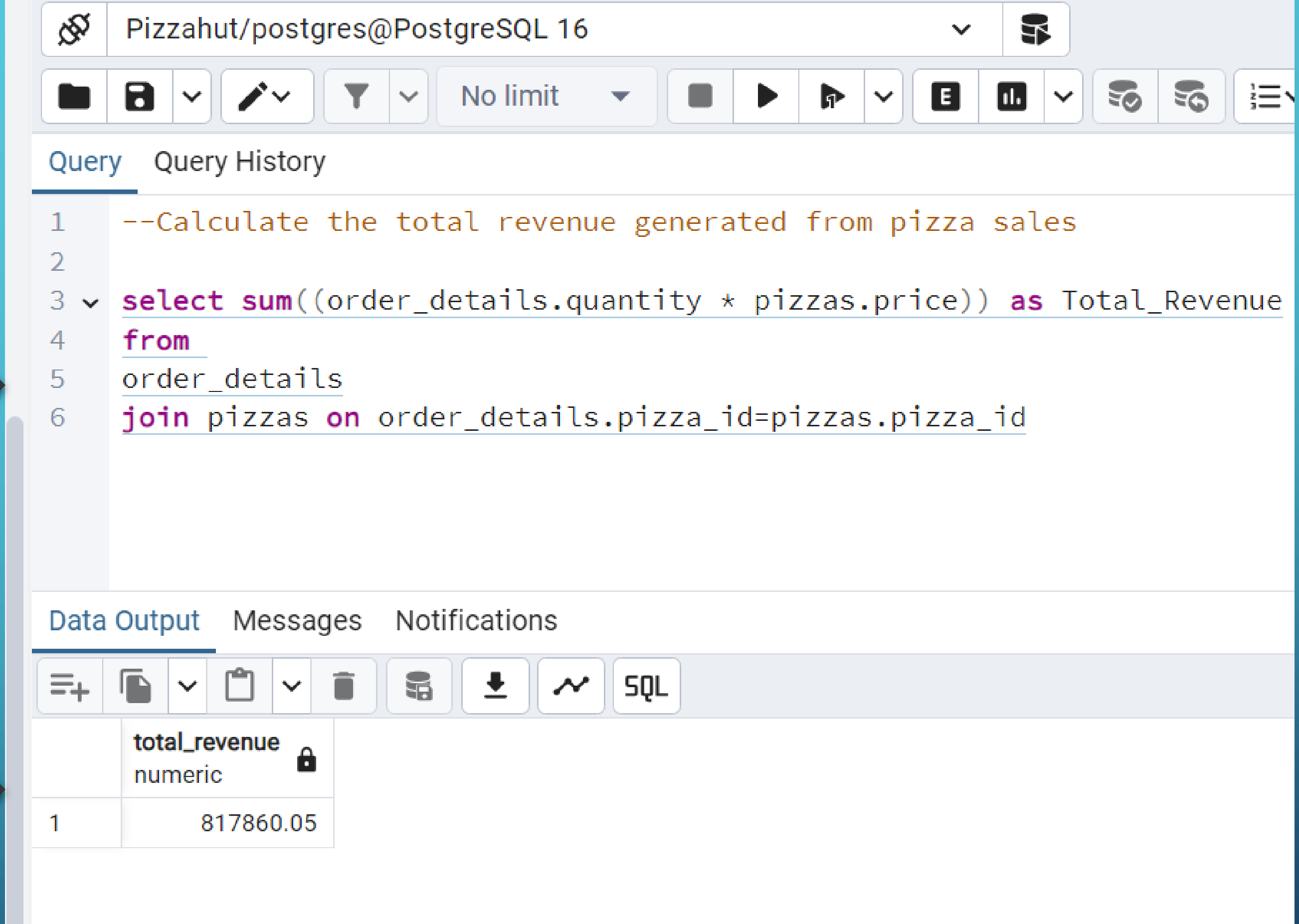


Pizzahut/postgres@PostgreSQL 16

No limit

```sql
--Retrieve the total number of orders placed

select count(*) from orders
```

	count bigint
1	21350



INPUT QUERY

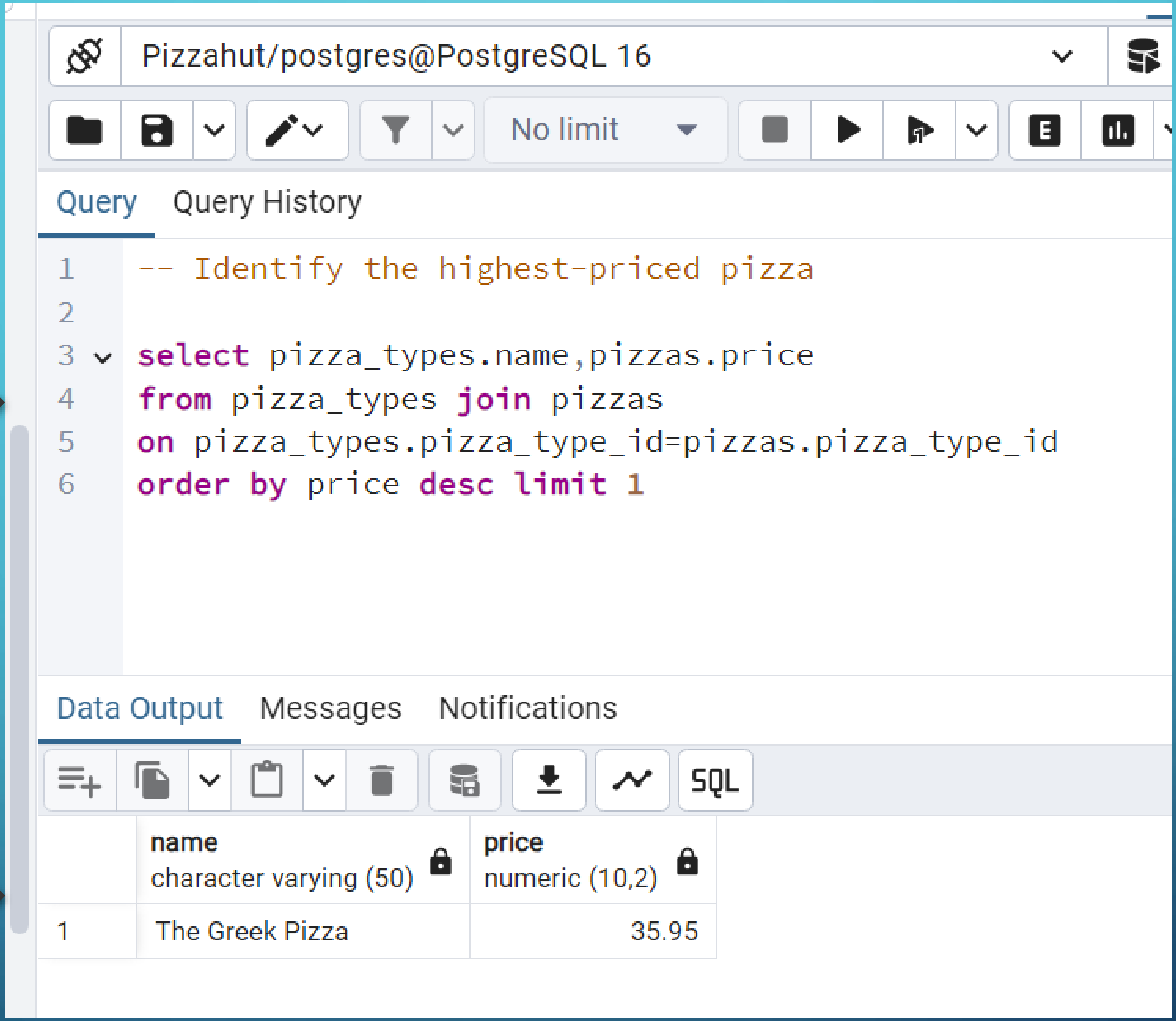
```sql
--Calculate the total revenue generated from pizza sales

select sum((order_details.quantity * pizzas.price)) as Total_Revenue
from
order_details
join pizzas on order_details.pizza_id=pizzas.pizza_id
```

OUTPUT

	total_revenue numeric
1	817860.05

INPUT QUERY

OUTPUT



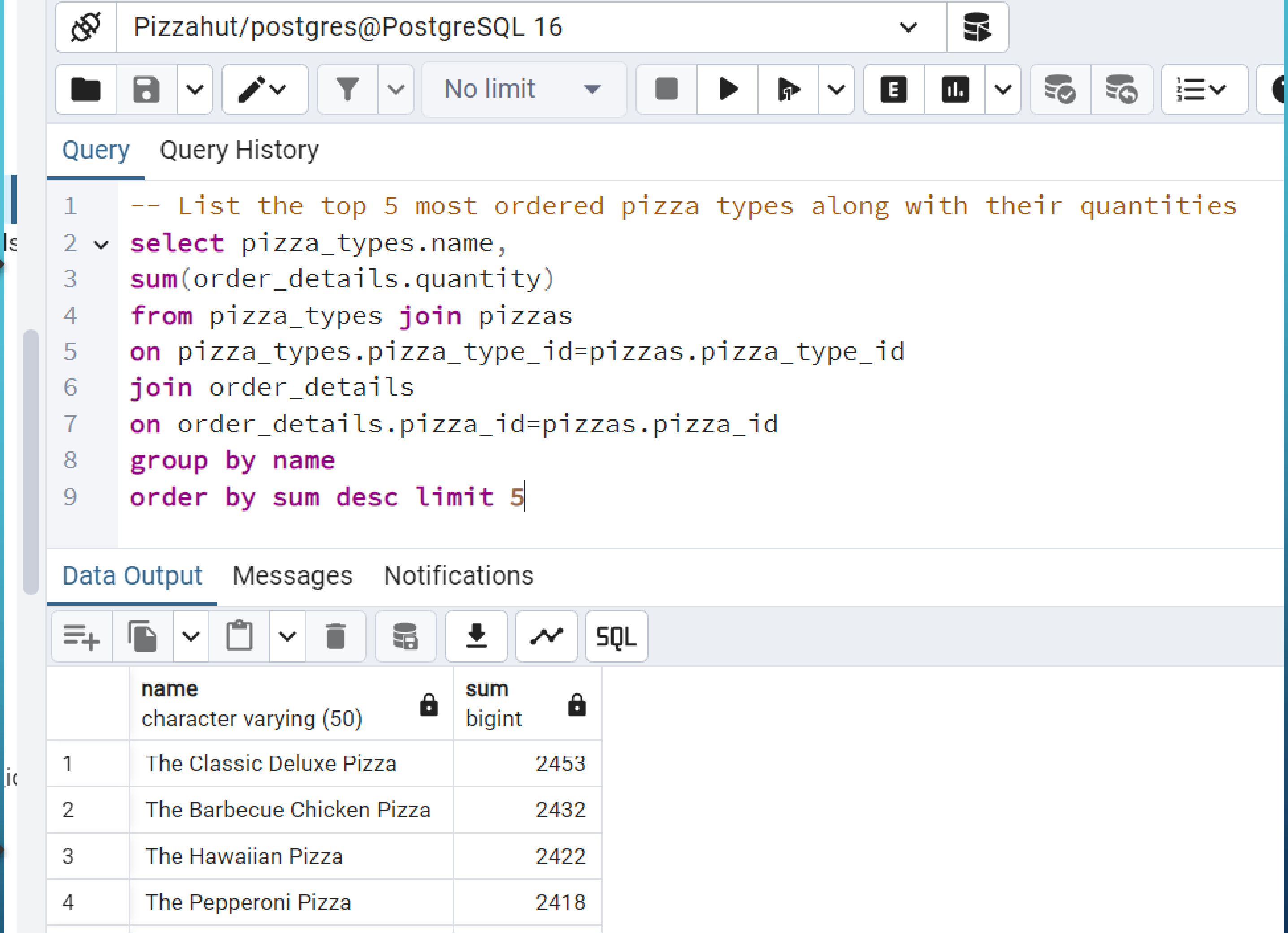
Pizzahut/postgres@PostgreSQL 16

```sql
-- Identify the highest-priced pizza

select pizza_types.name,pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
order by price desc limit 1
```

	name character varying (50)	price numeric (10,2)
1	The Greek Pizza	35.95

INPUT QUERY

OUTPUT

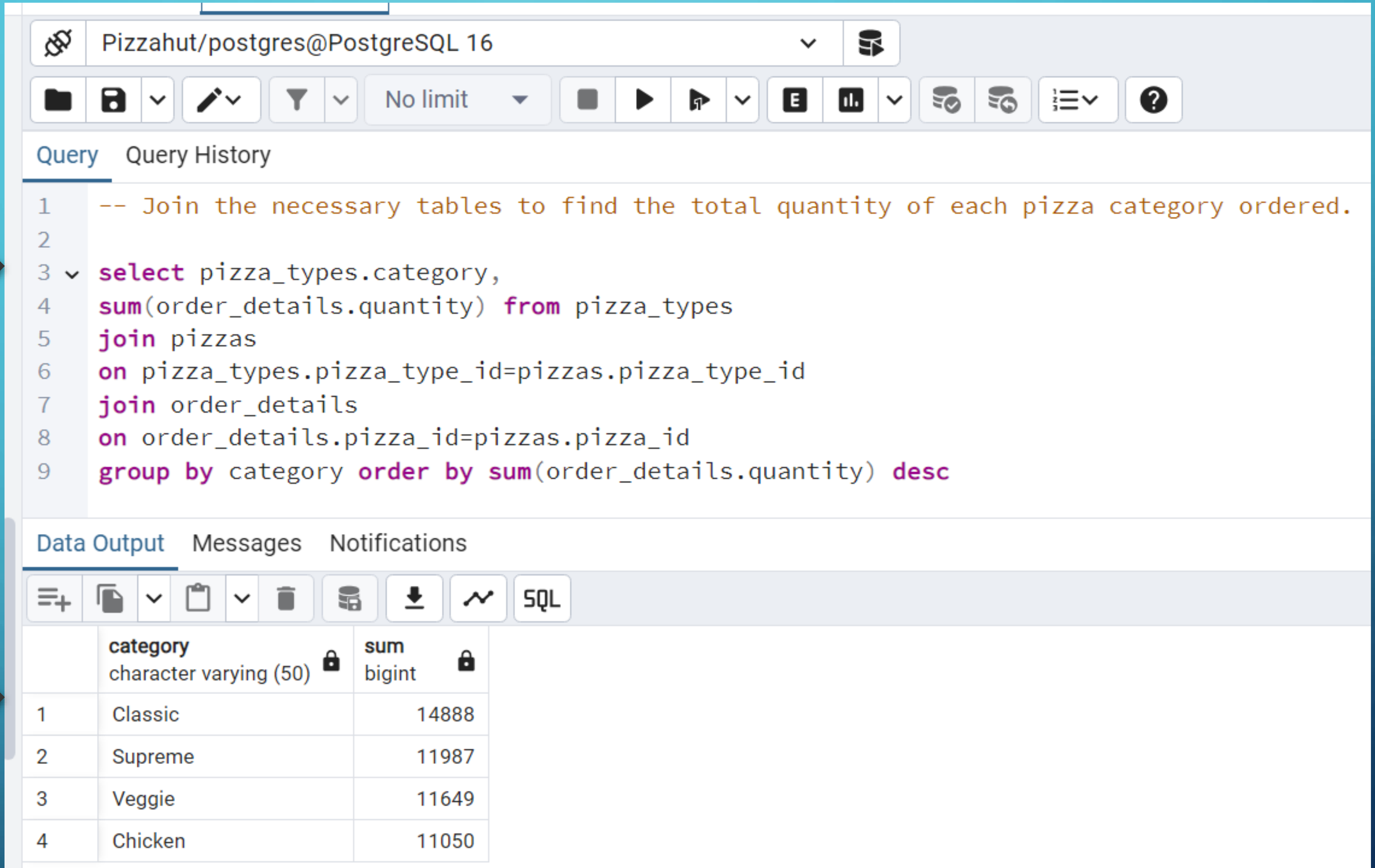


```sql
-- List the top 5 most ordered pizza types along with their quantities
select pizza_types.name,
sum(order_details.quantity)
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by name
order by sum desc limit 5
```

	name character varying (50)	sum bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418

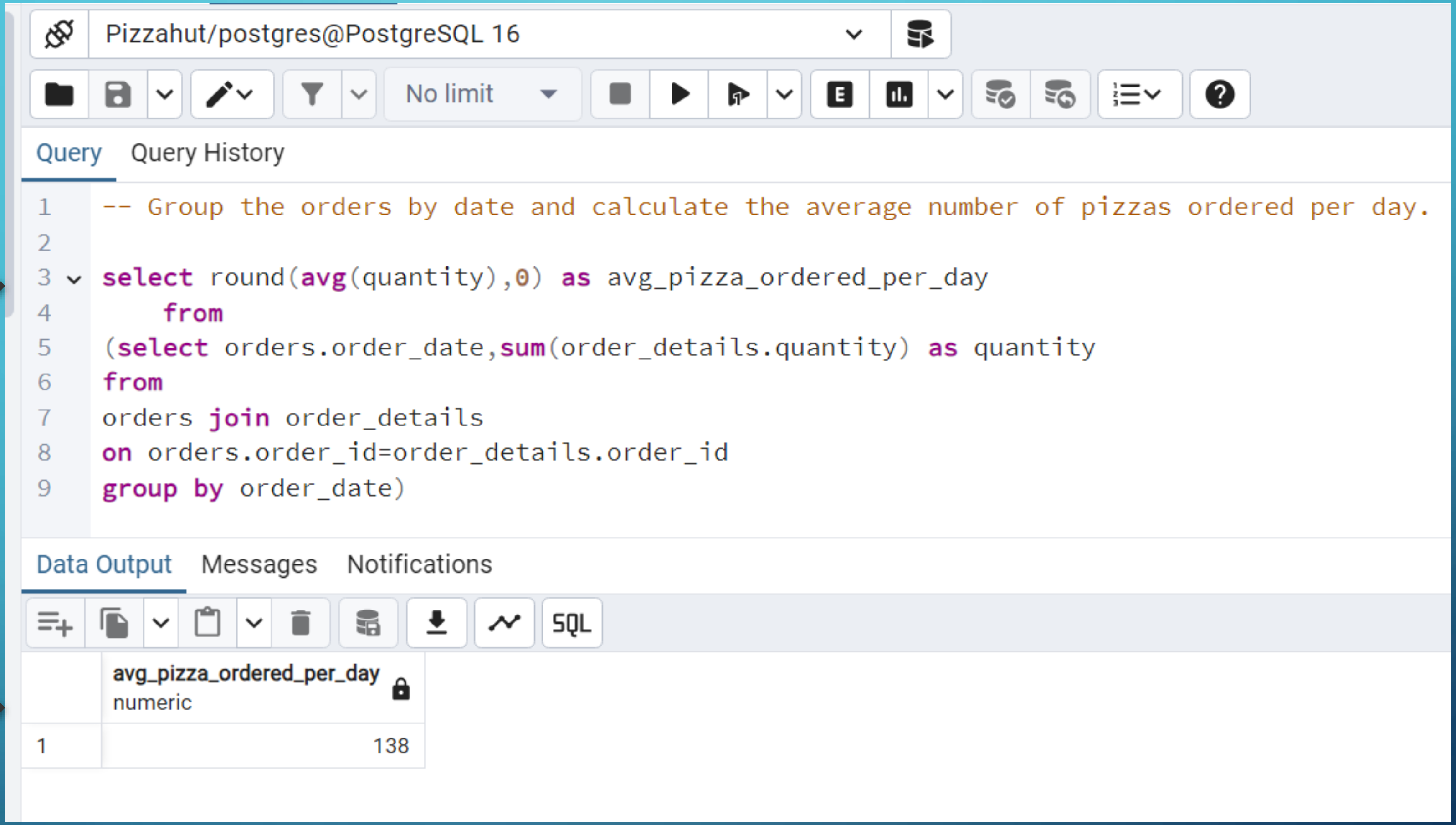
INPUT QUERY

OUTPUT



```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.

select pizza_types.category,
sum(order_details.quantity) from pizza_types
join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by category order by sum(order_details.quantity) desc
```

	category character varying (50)	sum bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050



Pizzahut/postgres@PostgreSQL 16

No limit

INPUT QUERY

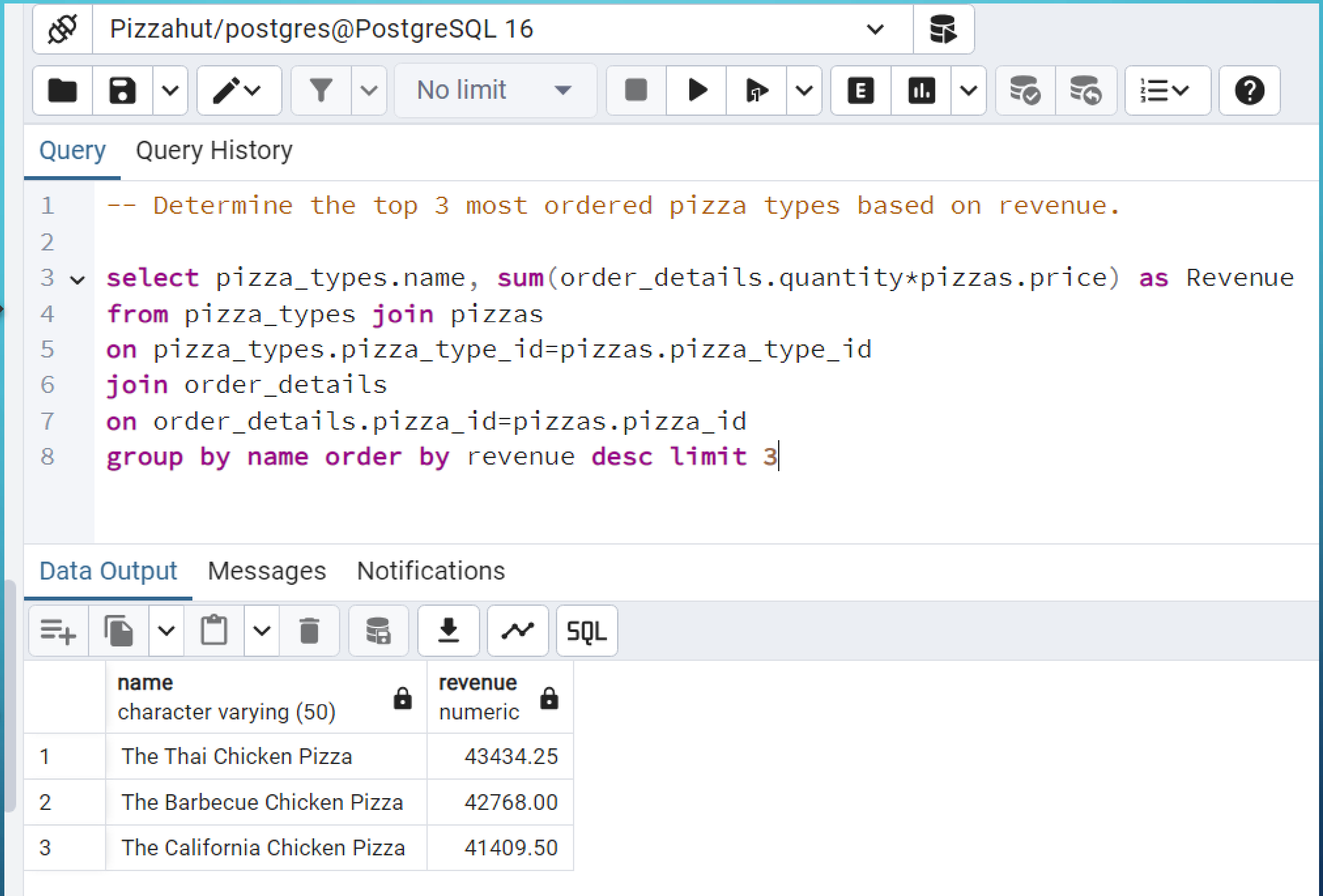
```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.

select round(avg(quantity),0) as avg_pizza_ordered_per_day
    from
(select orders.order_date,sum(order_details.quantity) as quantity
from
orders join order_details
on orders.order_id=order_details.order_id
group by order_date)
```

OUTPUT

```
   avg_pizza_ordered_per_day
   numeric
1                        138
```

INPUT QUERY

OUTPUT

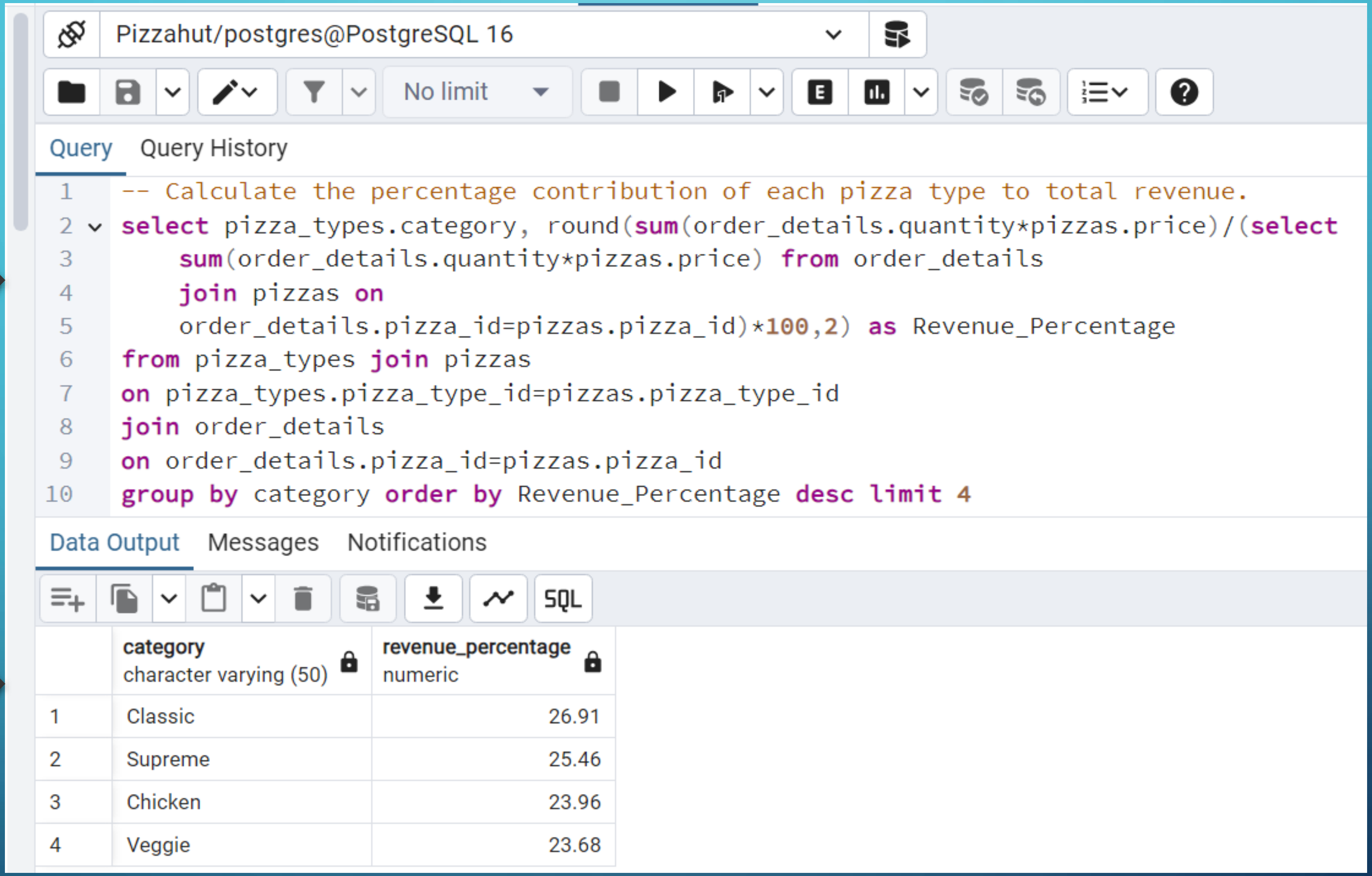


```sql
-- Determine the top 3 most ordered pizza types based on revenue.

select pizza_types.name, sum(order_details.quantity*pizzas.price) as Revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by name order by revenue desc limit 3
```

	name character varying (50)	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50

INPUT QUERY

OUTPUT



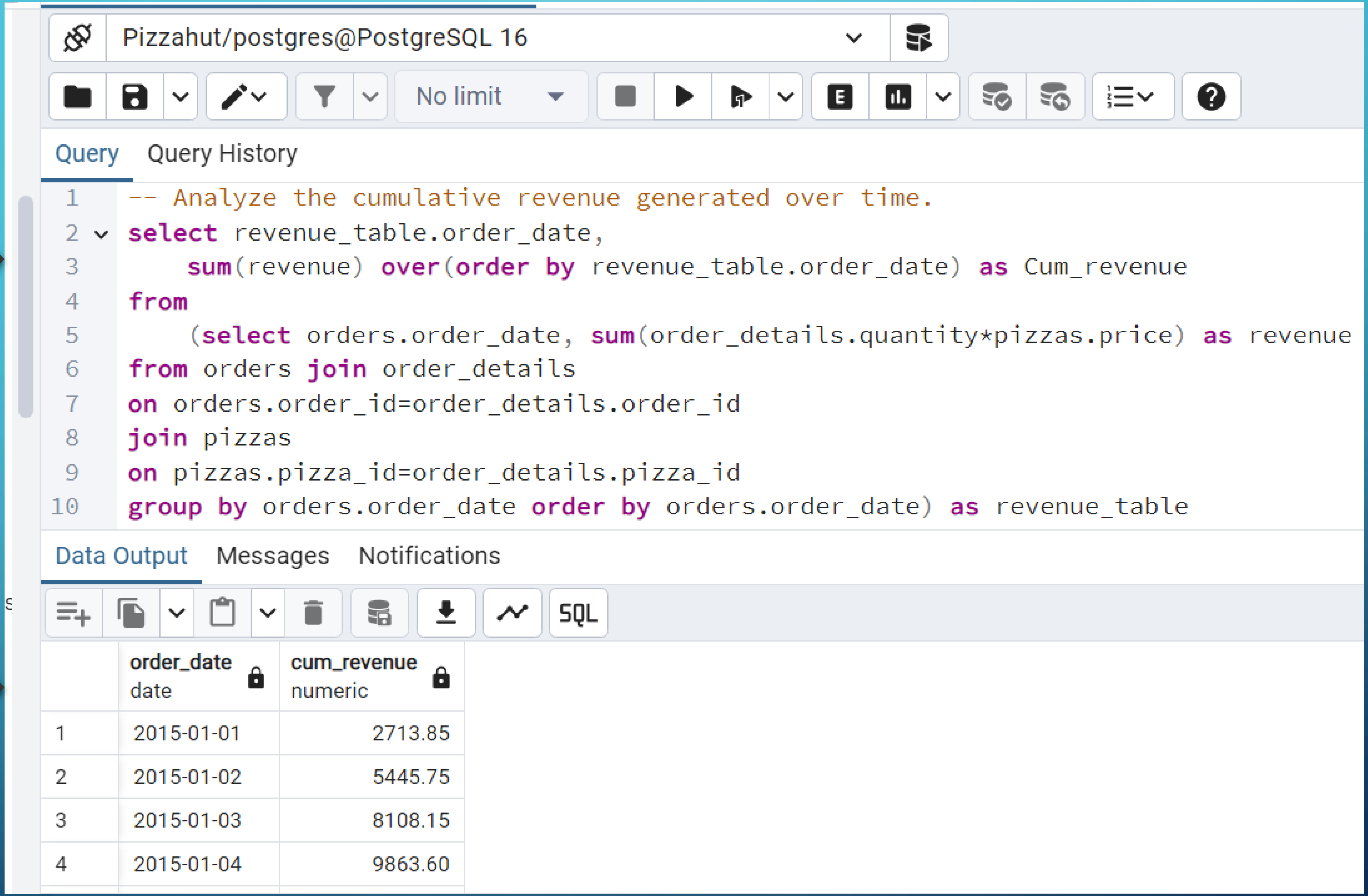
Pizzahut/postgres@PostgreSQL 16

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
select pizza_types.category, round(sum(order_details.quantity*pizzas.price)/(select
    sum(order_details.quantity*pizzas.price) from order_details
    join pizzas on
    order_details.pizza_id=pizzas.pizza_id)*100,2) as Revenue_Percentage
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by category order by Revenue_Percentage desc limit 4
```

	category character varying (50)	revenue_percentage numeric
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

INPUT QUERY

OUTPUT



Pizzahut/postgres@PostgreSQL 16

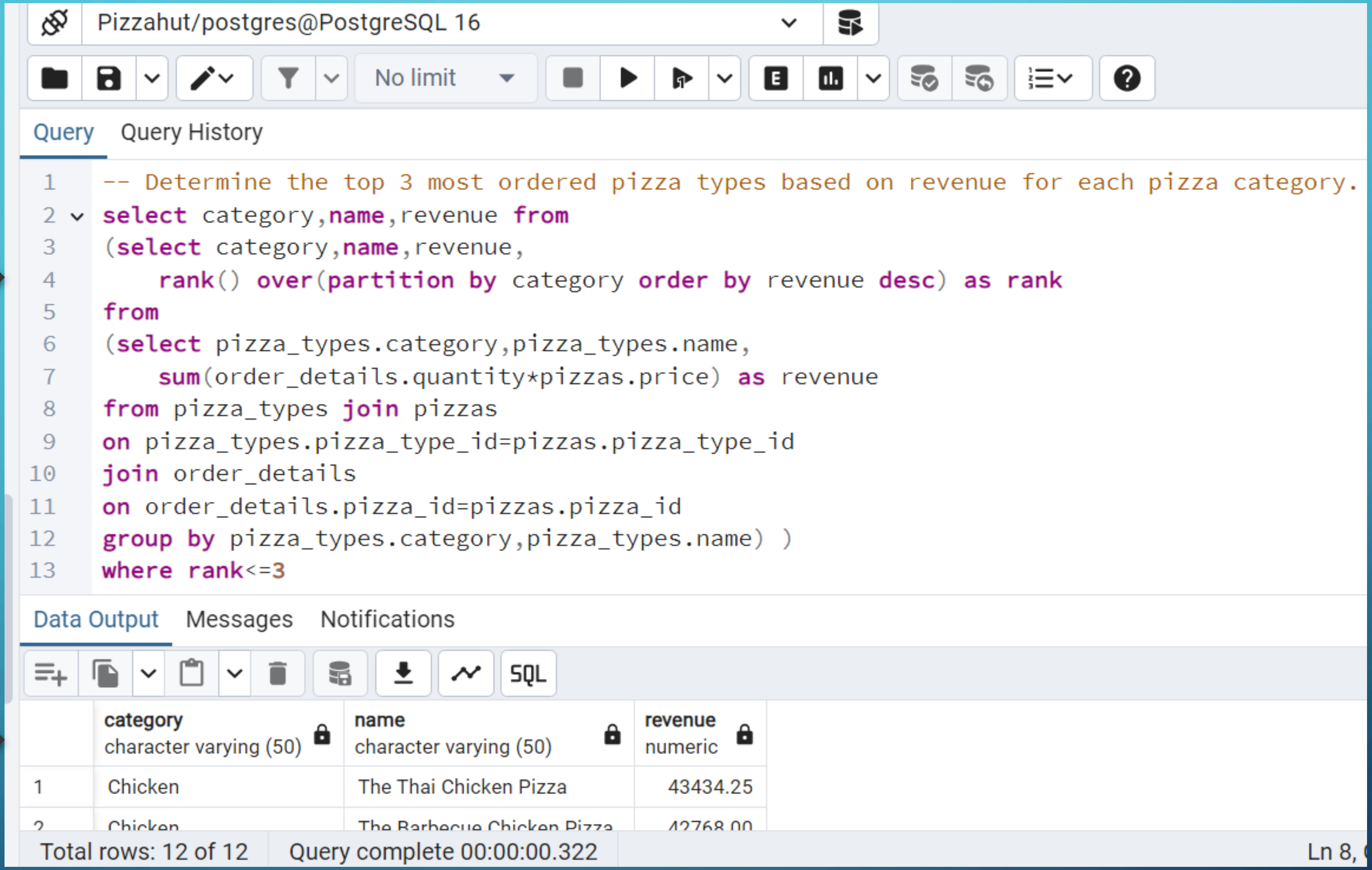
No limit

```
-- Analyze the cumulative revenue generated over time.
select revenue_table.order_date,
    sum(revenue) over(order by revenue_table.order_date) as Cum_revenue
from
    (select orders.order_date, sum(order_details.quantity*pizzas.price) as revenue
from orders join order_details
on orders.order_id=order_details.order_id
join pizzas
on pizzas.pizza_id=order_details.pizza_id
group by orders.order_date order by orders.order_date) as revenue_table
```

	order_date date	cum_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.60

INPUT QUERY

OUTPUT



```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
select category,name,revenue from
(select category,name,revenue,
    rank() over(partition by category order by revenue desc) as rank
from
(select pizza_types.category,pizza_types.name,
    sum(order_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.category,pizza_types.name) )
where rank<=3
```

	category character varying (50)	name character varying (50)	revenue numeric
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768.00

Total rows: 12 of 12   Query complete 00:00:00.322   Ln 8,